Rule-based modeling of transcriptional attenuation at the tryptophan operon

Celine Kuttler, Cédric Lhoussaine, Mirabelle Nebut

RULE-BASED MODELING OF TRANSCRIPTIONAL ATTENUATION AT THE TRYPTOPHAN OPERON

Céline Kuttler   Cédric Lhoussaine   Mirabelle Nebut

University of Lille 1
Bat M3, Cité Scientifique
59655 Villeneuve d'Ascq Cedex, FRANCE

Interdisciplinary Research Institute
BP 70478
59658 Villeneuve d'Ascq Cedex, FRANCE

ABSTRACT

Transcriptional attenuation at E.coli's tryptophan operon is a prime example of RNA-mediated gene regulation. In this paper, we present a discrete stochastic model of the fine-grained control of attenuation, based on chemical reactions. Stochastic simulation of our model confirms results that were previously obtained by master or differential equations. Our approach is easier to understand than master equations, although mathematically well founded. It is compact due to rule schemas that define finite sets of chemical reactions. Moreover, our model makes intense use of reaction rules with more than two reactants. As we show, such n-ary rules are essential to concisely capture the control of attenuation. Our model could not adequately be represented in object-centered modeling languages based on the pi-calculus, because these are limited to binary interactions.

1 INTRODUCTION

Transcriptional attenuation is a control mechanism in gene expression, that down-regulates transcription as a response to the increased speed of translation of regulatory codons within the nascent transcript, via alternative mRNA conformations. Although investigated within bacterial systems since the 1970s (Kasai 1974, Yanofsky 1981), the phenomenon attracted significantly less interest than the control of transcription initiation, mediated by DNA binding proteins. This dramatically changed in the 2000s with the discovery of regulatory mechanisms in higher organisms that exploit RNA properties. Quantitative investigations of regulation via RNA gained momentum for therapeutic approaches and synthetic biology (Barboric and Peterlin 2005, Beisel and Smolke 2009).

E.coli's tryptophan (*trp*) operon is the best understood biosynthetic operon. It consists in a sequence of jointly transcribed genes, that encode several enzymes. These allow the bacterium to synthesize the amino acid tryptophan, unless its environment provides sufficient amounts of it. Transcription attenuation mechanisms are common in biosynthetic operons across bacterial species (Gollnick et al. 2005, Gutierrez-Preciado et al. 2005, Yanofsky 2007). In *E.coli*, tryptophan regulation relies on two further mechanisms beyond attenuation.

Santillan and Zeron (2004) modeled all three levels of *trp* regulation in *E. coli* through delay differential equations (DDE), without investigating attenuation in detail. DDEs are usually directly derived from informal biochemical reactions. The main drawback of such deterministic models is that they only provide observations of average behavior. In particular, they do not account for possible stochastic noise from which multi-modal states may arise. In that case, the average behavior does not correspond to any of the actual states. Since regulatory systems involve few biological entities, a criterion known to increase stochastic effects, one may wonder if the deterministic assumption is appropriate regarding *E.coli*'s *trp* operon. This calls for stochastic modeling. The first stochastic treatment of attenuation at the *trp* operon indeed dates back to von Heijne, Nilsson, and Blomberg (1977).

Elf and Ehrenberg (2005) analyze the sensitivity of attenuation through probability functions and, more generally, discrete master equations. This approach benefits from a rich probability theory that gives valuable insights and measurement capabilities. However, apart from rare exceptions, master equations can only be evaluated numerically, and not solved symbolically. Each biological system requires an ad-hoc master equation or probability function that is usually hard to design from the mechanistic intuition of the system.

Discrete event models for stochastic simulation are typically described by chemical reactions. These can be studied within formal rule-based modeling languages (Chabrier-Rivier et al. 2005, Danos et al. 2007, Krivine et al. 2008, Ciocchetta and Hillston 2008). Molecular systems are understood as multisets of molecules that are rewritten by applying chemical reactions. Reaction speeds are derived from rate constants and cardinalities of sets. The stochastic semantics of chemical reactions is given in terms of continuous time Markov chains (CTMCs). The algorithm of Gillespie (1976) allows direct stochastic simulation, starting from a given set of molecules and a set of chemical reactions. Rule-based models are intuitive in the sense that they describe molecular interactions and are simpler to modify and extend than models based on classical mathematical functions.

In this work, we present the first formal rule-based stochastic model of attenuation at *E. coli*'s tryptophan operon. Our representation addresses the same level of detail as the work of Elf and Ehrenberg (2005), moreover it follows the tradition of stochastic models of gene expression of Arkin, Ross, and McAdams (1998). Our model consists of 71 chemical reactions, that are generated from 13 rule schemas. These faithfully summarize the rich narrative account in the biological literature (Trun and Trempy 2003, Konan and Yanofsky 1999, Gollnick 2004). We obtain this concise description by two ingredients, rules schemas (introduced in Section 3) and n-ary chemical reactions. Rule schemas allow to represent finite sets of chemical reactions in a compact manner, which differ only in the choice of some molecule parameters e.g. folding or binding state, location, etc.

As our work indicates, n-ary reactions are indispensable to intelligible representations of the *trp* attenuator. We prospect the same holds for many other cases. By n-ary reactions, we refer to rules with three or more inputs. Certain authors as Cardelli and Zavattaro (2008) and Pradalier et al. (2007) support the idea that binary reactions are sufficient to represent chemical knowledge. They do so to advocate formal *object-centric* representations that are confined to binary interactions, namely recent languages based on the stochastic π-calculus (Regev 2002, Phillips and Cardelli 2007, Kuttler et al. 2007, Dematté et al. 2008). However, rewriting n-ary to binary reactions is tedious and requires sufficient expressiveness of formal languages. Sequences of reactions need to be executed within atomic *transactions*, so that no other interactions intervene. The benefit of n-ary reactions to modeling is that they allow to incorporate a global control. Our model deploys three categories of global control: conditions for the application of binary rules, side effect resulting from the interaction between two actors on others, and switching between abstraction levels.

We review the biological background in Section 2, and introduce our rule-based language in Section 3. As a first example, we model the multi-step race between transcription and translation by four rule schemas in Section 4. We quantitatively investigate the hyper-sensitivity of this simplified attenuation model by simulation with the Kappa Factory (Danos et al. 2007). In Section 5, we present our model of attenuation at *E.coli*'s *trp* operon. We qualitatively reproduce and confirm results of Elf and Ehrenberg (2005) regarding the probability of transcription as a function of the rate of trp-codon translation.

2   TRANSCRIPTIONAL ATTENUATION

In the following, we review ribosome-mediated transcriptional attenuation, a regulatory mechanism used by many bacteria, and detail on the case of E.coli's tryptophan (*trp*) operon (Yanofsky 1981, Gollnick 2004, Yanofsky 2000). Note that we only cover as much detail as will be incorporated into our formal model, and omit other aspects that would be covered in the biological literature (e.g. the role of transfer RNA in translation, which are not explicitly rendered by our model).

The role of transcriptional attenuation is to prematurely interrupt the transcription of a DNA sequence, in order to avoid the production of the proteins encoded by the sequence, in cases where these proteins are not required. The proteins encoded by E.coli's *trp* operon are enzymes for the biosynthesis of the amino acid tryptophan (Trp), that the bacterium only needs if its food doesn't provide sufficient amounts of tryptophan.

We first sketch the attenuation mechanism, before going into detail. Attenuation is a dynamical process that involves both transcription and translation. It is controlled by mRNA secondary structures, or *hairpins*, into which the initial portion (or: *leader*) of the *trp* transcript folds. The so-called *terminator* hairpin aborts transcription, see Figure 1 (A). Hairpin formation is determined by a *race* between the transcribing polymerase and the translating ribosome. Two *trp* codons within the leader are crucial. If tryptophan is abundant, the ribosome wins the race, the terminator hairpin forms, and transcription aborts, producing only the truncated leader mRNA. If conversely tryptophan is rare, the ribosome slows down, the anti-terminator forms, and transcription continues into the full operon.

We now recall the basic processes of transcription and translation, when attenuation is not involved. Transcription is the first phase of gene expression that copies information content from DNA into mRNA. It begins with the binding of an RNA polymerase (RNAP) to a dedicated start sequence on DNA. RNAP starts assembling an mRNA sequence, advancing stepwise over DNA at an average rate of 50 nucleotides per second. In each step, it extends the growing mRNA sequence by one new nucleotide. Translation translates mRNA into sequences of amino acids (that later fold into proteins). It starts
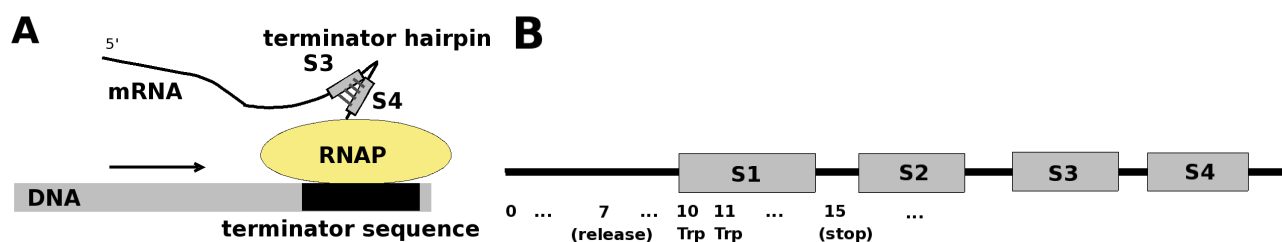
Figure 1: **(A)** Transcription terminates when a hairpin forms in the most recent portion of mRNA, while RNAP is paused on the terminator DNA sequence. **(B)** Control region of the *trp* mRNA. Adjacent pairs of the four segments fold into alternative hairpins. The *terminator* hairpin $S_3 \cdot S_4$ prematurely aborts transcription of the *trp* operon, while the *anti-terminator* $S_2 \cdot S_3$ allows transcription to continue into the full operon.

with the binding of a ribosome to the free end of an mRNA, the other end of which is still being elongated by RNAP. The ribosome advances by *codons*, words of three mRNA nucleotides, for each of which it adds one amino acid to the growing sequence. While the average rate of translation is 15 codons per second, each step of the ribosome is actually limited by the abundance of the currently required amino acid. The ribosome slows down on codons for which the corresponding amino acid is in short supply. It thus acts as a sensor for the concentration of that amino acid.

We now detail the attenuation mechanism at E.coli's *trp* operon. It exploits the organization of the *leader* part of the operon, which are the first 150 of several thousand nucleotides. The corresponding mRNA comprises four segments S_1 to S_4, illustrated by Figure 1 (B). Two adjacent segments fold into a hairpin, unless either of them is occupied by a ribosome. We distinguish three different secondary structures of the *trp* leader mRNA, named by their respective roles in attenuation. The hairpin $S_1 \cdot S_2$ is referred to as the *pause* hairpin, $S_2 \cdot S_3$ is the *anti-terminator*, and $S_3 \cdot S_4$ the *terminator*. Each segment can at most participate to one hairpin at a time. Most importantly, the anti-terminator prohibits the formation of the terminator hairpin by sequestering S_3; it also excludes the pause hairpin (because both require S_2). On the other hand, the pause hairpin ($S_1 \cdot S_2$) and the terminator ($S_3 \cdot S_4$) can co-occur. Note that hairpins form on a faster time scale than any other reaction in the system.

A translating ribosome disrupts a hairpin along its way, without significantly slowing down. The impact of hairpins on transcription is fundamentally different. After RNAP has transcribed the segments S_1 and S_2, the pause hairpin has formed, RNAP remains stalled on a strong DNA *pause site* until the mRNA hairpin is opened by a ribosome. This sets the starting signal for the attenuation race.

The DNA leader contains a *terminator sequence*, just after the portion that encodes S_4, see Figure 1 (A). When RNAP reaches this terminator sequence, the terminator mRNA hairpin forms – unless the anti-terminator hairpin has already appeared. The combination of terminator mRNA hairpin and terminator DNA sequence aborts transcription. In presence of the anti-terminator hairpin however, RNAP transcribes through the terminator sequence and continues into the full operon.

We are now at the core of this section. Just before the attenuation race starts, RNAP is paused on a nucleotide that we refer to as DNA_0. The approaching ribosome disrupts the pause hairpin when stepping onto the 7th codon. This is the starting signal for the race, transcription and translation are now synchronized. The mRNA segment S_3 becomes available when RNAP reaches the 36th nucleotide, S_4 follows at nucleotide DNA_{47}.

The decisive point of the attenuation race are the two *trp* codons at positions 10 and 11 of the leader mRNA. These *regulatory* or *control* codons act as sensors for the concentration of tryptophan. They determine the speed of the ribosome's forward movement. If tryptophan is in short supply, the ribosome stalls on codons 10 or 11. As Figure 1 (B) shows, the ribosome then remains within S_1. The anti-terminator hairpin can then form from the segments S_2 and S_3, which prevents formation of the terminator, and allows transcription to continue. This is depicted as *readthrough configuration* in Figure 2.

Conversely if tryptophan is abundant, the ribosome efficiently translates the control codons. As soon as it reaches the 13th codon, its footprint partially covers segment S_2, which renders anti-terminator formation impossible. The ribosome eventually reaches the stop codon at position 15, from where it unbinds after approximately one second. This delay is considerable, compared to the other reactions in the system. Meanwhile RNAP continues transcription, producing S_3 and S_4. The terminator hairpin forms and transcription aborts, before the ribosome dissociates. This is depicted as *termination configuration* in Figure 2.

The third possible outcome is not covered by Figure 2. The ribosome may dissociate from the stop codon after S_3 has been transcribed, but before S_4 becomes available. In this case S_1, S_2 and S_3 are available, and either the pause or the
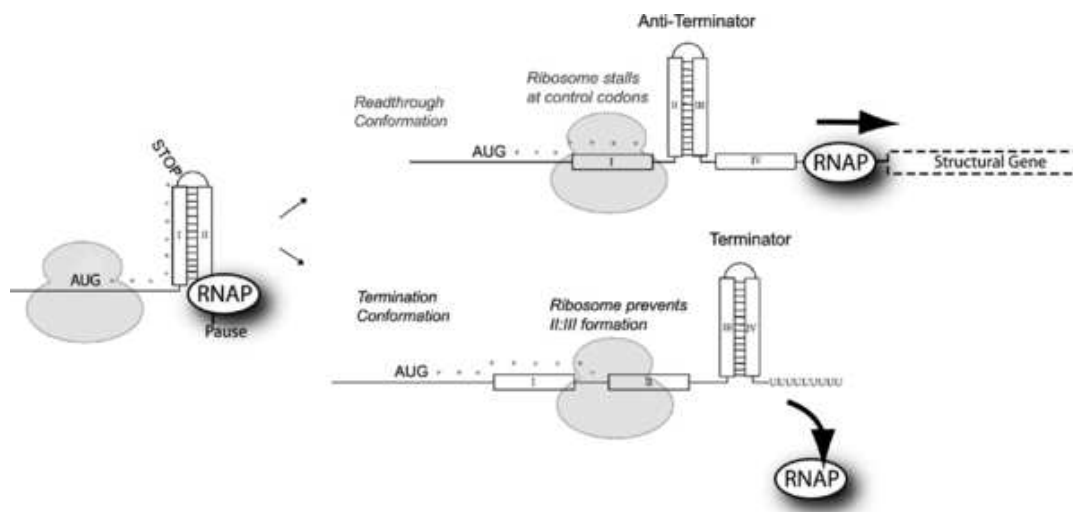
Figure 2: Starting point and possible outcomes of the race at *E.coli*'s *trp* operon. Left: RNAP is paused by the pause hairpin, awaiting to be released by the ribosome's next step. Upper right: when Trp supply is low, the ribosome stalls at the control codons 10 and 11, the anti-terminator hairpin forms and transcription continues into the operon. Lower right: when Trp supply is high, the ribosome rapidly translates over the control codons. Before it unbinds from the stop codon, the terminator hairpin forms and transcription aborts. Figure reproduced with permission from Elf and Ehrenberg, 2005.

anti-terminator forms, with equal probability but in mutual exclusion. The anti-terminator allows transcription to continue despite of high tryptophan supply. This sets the *basal read-through* level of the operon, that is experimentally estimated to $10 - 15\%$ (JR Roesser and Yanofsky 1989).

3 RULE SCHEMAS FOR CHEMICAL REACTIONS

In this section we provide a formal and minimal rule-based language tailored to our needs. Note that this language is not intended to be a proper contribution. We define chemical reactions, that operate on multisets of complex molecules with attributes such as $\text{RNAP} \cdot \text{DNA}(23)$. The attribute value 23 here is the location of the DNA nucleotide, to which RNAP is bound. Other attributes of interest can be the compartment of a molecule, or information on its states, for instance folding or phosphorilation.

We then present a language of rules schemas allowing to define finite sets of chemical reactions in a compact manner. Rule schemas are like chemical reactions, except that attribute values are now extended to expressions with variables. Complex molecules are thus described by terms such as $\text{RNAP} \cdot \text{DNA}(x+1)$ where x is a variable with values in $\{1, \dots, 51\}$. All variables are universally quantified over finite sets, such that they define finite sets of reactions.

3.1 Chemical Reactions

In order to define the syntax of attributed molecules, we fix a possibly infinite set of attribute values $\mathscr{C}$ and a finite set $\mathscr{N}$ of molecule names. We assume that every molecule name $N \in \mathscr{N}$ has a fixed arity $ar(N) \geq 0$ which specifies the number of attributes of N.

A *molecule M*, defined in Table 1, is a complex of attributed molecules. We write $M_1 \cdot M_2$ for the complex of M_1 and M_2. For instance, if $\text{RNAP}, \text{DNA} \in \mathscr{N}$ and $47 \in \mathscr{C}$ then $\text{RNAP} \cdot \text{DNA}(47)$ is molecule complex consisting of an RNAP that is bound to the DNA nucleotide at position 47. A *chemical solution S* is a multiset union of molecules. A *chemical reaction* is a rule that rewrites a solution S_1 into a solution S_2 assigned with a stochastic rate constant $k \in \mathbb{R}^+$. For instance, the following reaction states that an RNAP bound to the DNA nucleotide at position 23 may move forwards to the DNA nucleotide at position 24. The speed of this reaction is 50 sec^{-1}:
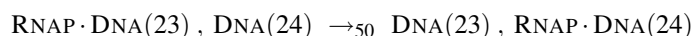
$$\text{RNAP} \cdot \text{DNA}(23), \ \text{DNA}(24) \ \rightarrow_{50} \ \text{DNA}(23), \ \text{RNAP} \cdot \text{DNA}(24)$$

Table 1: Chemical reactions where $N \in \mathcal{N}$, $c_1, \ldots, c_n \in \mathscr{C}$, $ar(N) = n$ and $k \in \mathbb{R}^+$.

Molecules	$M \in Mol$	$::=$	$N(c_1, \ldots, c_n) \mid M_1 \cdot M_2$
Solutions	$S \in Sol$	$::=$	$M \mid S_1, S_2$
Reactions	$S_1 \rightarrow_k S_2$		

Table 2: Rule schemas where $x \in \mathcal{V}$, $c \in \mathscr{C}$, $f \in \mathscr{F}$, $e_1, \ldots, e_n \in Exp$, $N \in \mathcal{N}$, $ar(N) = n$, $D_1, \ldots, D_n \subseteq \mathscr{C}$ are finite sets, and $k \in \mathbb{R}^+$.

Expressions	$e \in Exp$	$::=$	$x \mid c \mid f(e_1, \ldots, e_n)$
Schematic molecules	$M \in SMol$	$::=$	$N(e_1, \ldots, e_n) \mid M_1 \cdot M_2$
Schematic solution	$S \in SSol$	$::=$	$M \mid S_1, S_2$
Rule schema	$\forall x_1 \in D_1 \ldots \forall x_n \in D_n.\ S_1 \rightarrow_k S_2$		where $\mathcal{V}(S_1) \cup \mathcal{V}(S_2) \subseteq \{x_1, \ldots, x_n\}$

Of course, one needs many similar rules for the many other DNA nucleotides with different positions. We next introduce rule schemas, by which to define such sets of chemical reactions in a compact manner.

3.2 Rule Schemas

In order to define rule schemas for chemical reactions, we need variables x for attribute values and expressions such as $x+1$, in order to compute corresponding attribute values. This permits to generalize the above chemical reaction to the following rule schema:

$$\forall x \in \{1, \ldots, 49\}.\ \text{RNAP} \cdot \text{DNA}(x),\ \text{DNA}(x+1) \quad \rightarrow_{50} \quad \text{DNA}(x),\ \text{RNAP} \cdot \text{DNA}(x+1) \tag{1}$$

We thus need a set $\mathcal{V}$ of variables that are ranged over by x, and a finite set $\mathscr{F}$ of function symbols $f \in \mathscr{F}$ with arities $ar(f) \geq 0$. Furthermore, we assume an interpretation $[\![f]\!] : \mathscr{C}^{ar(f)} \rightarrow \mathscr{C}$ for every $f \in \mathscr{F}$. An *expression e* with values in $\mathscr{C}$ is a term with the abstract syntax given in Table 2. In our modeling case studies, we will assume that symbol $+ \in \mathscr{F}$ of arity 2 is interpreted as addition on natural numbers. We freely use infix syntax as usual, i.e. we write $e_1 + e_2$ instead of $+(e_1, e_2)$. Given a variable assignment $\alpha : \mathcal{V} \rightarrow \mathscr{C}$, every expression $e \in Exp$ denotes an element $[\![e]\!]_\alpha \in \mathscr{C}$ that we define as follows:

$$[\![c]\!]_\alpha = c \qquad [\![x]\!]_\alpha = \alpha(x) \qquad [\![f(e_1, \ldots, e_n)]\!]_\alpha = [\![f]\!]([\![e_1]\!]_\alpha, \ldots, [\![e_n]\!]_\alpha)$$

A *schematic molecule M* is like a molecule, except we now allow for expressions in attribute positions rather than attribute values only. A *schematic solution* $S \in SSol$ is a multiset of schematic molecules. As usual, we write $\mathcal{V}(S)$ for the set of variables that occur in molecules of S. A *rule schema* specifies the domains of variables occurring in the schematic solutions of the rule.

For every variable assignment $\alpha : \mathcal{V} \rightarrow \mathscr{C}$ that maps variables to values in their domain, we can instantiate the rule schema to finitely many reactions. A schematic molecule M is mapped to a molecule $[\![M]\!]_\alpha \in Mol$. Similarly, schematic solutions $S \in SSol$ get instantiated to solutions $[\![S]\!]_\alpha \in Sol$:

$$[\![N(e_1, \ldots, e_n)]\!]_\alpha = N([\![e_1]\!]_\alpha, \ldots, [\![e_n]\!]_\alpha) \qquad [\![M_1 \cdot M_2]\!]_\alpha = [\![M_1]\!]_\alpha \cdot [\![M_2]\!]_\alpha \qquad [\![S_1, S_2]\!]_\alpha = [\![S_1]\!]_\alpha, [\![S_2]\!]_\alpha$$

A rule schema is instantiated to a set of chemical reactions, by enumerating the chemical reactions for all variables assignments licensed by the quantifiers:

$$[\![\forall x_1 \in D_1 \ldots \forall x_n \in D_n.\ S_1 \rightarrow_k S_2]\!] = \{[\![S_1]\!]_\alpha \rightarrow_k [\![S_2]\!]_\alpha \mid \alpha : \mathcal{V} \rightarrow \mathscr{C},\ \alpha(x_1) \in D_1, \ldots, \alpha(x_n) \in D_n\}$$

3.3 Stochastic Semantics And Simulation

For the sake of completeness, we recall the stochastic semantics of chemical reactions and how to use them for stochastic semantics with Gillespie's algorithm. This underlines that our biological modeling case studies are indeed expressed in a formal modeling language.

The semantics of a set of chemical reactions is a continuous time Markov chain (CTMC). The states of the CTMCs are congruence classes $[S]_{\equiv}$ of chemical solutions S with respect to the least congruence relation $\equiv$ that makes complexation and summation associative and commutative:

$$M_1 \cdot M_2 \equiv M_2 \cdot M_1 \qquad (M_1 \cdot M_2) \cdot M_3 \equiv M_1 \cdot (M_2 \cdot M_3) \qquad S_1 , S_2 \equiv S_2 , S_1 \qquad (S_1 , S_2) , S_3 \equiv S_1 , (S_2 , S_3)$$

We introduce transitions $S \xrightarrow[L]{k} S'$ stating that S can be reduced to S' by applying a chemical reaction with rate constant $k \in \mathbb{R}^+$ to the subset of molecules in S with positions in L. Positions are the indexes in multisets such as $M_1 , \dots , M_n$ that we also write as $\oplus_{i=1}^n M_i$. We next introduce transitions $S \xrightarrow{r} S'$, where r sums up all rate constants of chemical reactions reducing S to S', as many times as they apply for some index set L.

$$\frac{L \subseteq \{1,\dots,n\} \qquad \oplus_{i \in L} M_i \equiv S \qquad S \to_k S'}{\oplus_{i=1}^n M_i \xrightarrow[L]{k} S' , \oplus_{i \notin L} M_i} \qquad\qquad \frac{r = \sum_{\{(L,k)|S \xrightarrow[L]{k} S_1 \equiv S'\}} k}{S \xrightarrow{r} S'}$$

Such translations are invariant under structural congruence, i.e. for all $S \equiv S_1$ and $S' \equiv S_1'$ it holds that $S \xrightarrow{r} S'$ if and only if $S_1 \xrightarrow{r} S_1'$. We can thus define $[S]_{\equiv} \xrightarrow{r} [S']_{\equiv}$ by $S \xrightarrow{r} S'$ as the transitions of the CTMC.

Gillespie's algorithm for stochastic simulation takes as input a finite set of chemical reactions and a chemical solution S. It then computes the overall rate of all possible transitions $R = \sum_{\{(r,[S_1]_{\equiv})|S \xrightarrow{r} S_1\}} r$, selects with probability r/R a congruence class $[S_1]_{\equiv}$ with transition $S \xrightarrow{r} S_1$, and returns an arbitrary representative of this congruence class $S' \equiv S_1$ jointly with a time delay drawn randomly from the exponential distribution with rate r.

3.4 Expressiveness And Related Rule-Based Languages

All models using rule schemas can be compiled, by instantiation, to finite collections of simple and, formally well-defined, chemical reactions. Although reactions do not define a Turing-complete language (Cardelli and Zavattaro 2008), their expressiveness is sufficient for our purpose. Furthermore, such collections of reactions are supported by standard tools for stochastic simulation such as Dizzy (Ramsey et al. 2005) or the rule-based language BioCham (Chabrier-Rivier et al. 2005).

Using rule schemas rather than directly reactions leads to compact models, thus increasing their readability. Note that, even if rule schemas could be defined solely by means of variables, function symbols allow a better control and precision on the collection of reactions that are generated. For example, without function symbols, we would have to represent DNA sequences by means of *name sharing*. Each DNA nucleotide would bear two parameters: one refers to its predecessor, the other to its successor. Given link names $\{\ell_0,\dots,\ell_{50}\}$, our previous rule (1) reads as

$$\forall x,y,z \in \{\ell_0,\dots,\ell_{50}\}. \; \text{RNAP} \cdot \text{DNA}(x,y) , \text{DNA}(y,z) \to_{50} \text{DNA}(x,y) , \text{RNAP} \cdot \text{DNA}(y,z)$$

Then starting from a DNA sequence $\text{DNA}(\ell_0,\ell_1),\text{DNA}(\ell_1,\ell_2),\dots,\text{DNA}(\ell_{49},\ell_{50})$, this rule schema instantiates into more ground rules than needed. For example, the rule $\text{RNAP} \cdot \text{DNA}(\ell_1,\ell_{10}) , \text{DNA}(\ell_{10},\ell_{45}) \to_{50} \text{DNA}(\ell_1,\ell_{10}) , \text{RNAP} \cdot \text{DNA}(\ell_{10},\ell_{45})$ is a meaningless instance of the above schema.

Alternative rule-based languages with higher expressiveness are Turing complete, e.g. the graph rewriting language Kappa (Danos et al. 2007), BioNetGen (Hlavacek et al. 2006), and the bigraphs of Krivine et al. (2008). Their pattern based graph rewriting rules resemble schemas, in that they can be applied to arbitrary subgraphs satisfying the pattern. In contrast to our approach, such schemas may describe infinitely many reactions. Furthermore, stochastic simulation is possible without inferring all those reactions on before hand. This generation process in uncritical in the present paper, since the overall number of reactions remains small, but is the bottleneck in other applications (Danos et al. 2007), where it grows exponentially. These formal rule-based languages were designed and used so far to tackle protein-protein interactions that
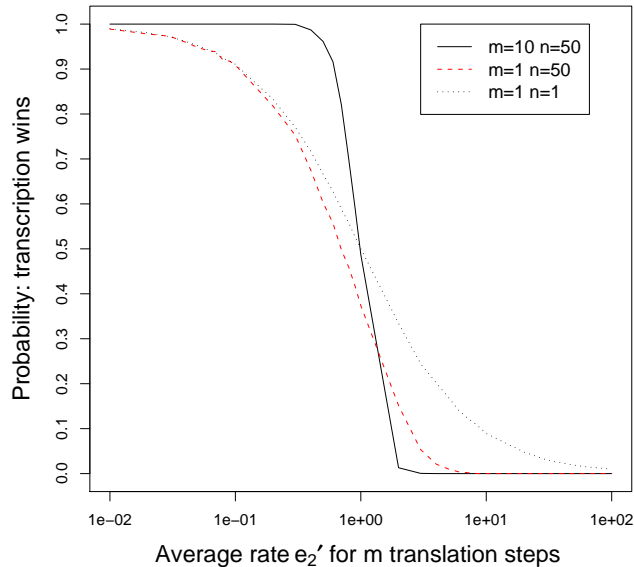
Figure 3: Probability that transcription wins, as a function of the average translation rate e_2', for different numbers of translation (m) versus transcription (n) steps.

occur in cellular signaling such as metabolic pathways. In contrast to this, our rule-based model deals with a fine-grained mechanism of gene regulation.

4 EXAMPLE: MULTI-STEP RACE

In this section we illustrate rule schemas for chemical reactions with a simple yet interesting example, borrowed from Elf and Ehrenberg (2005). Abstracting away from its detailed control conditions (see Section 2), transcriptional attenuation boils down to a race between the two competing multi-step processes of transcription and translation. As intuition easily confirms, the probability that transcription wins the race drops when the ribosome speeds up, and vice versa. Elf and Ehrenberg refined this insight. As they demonstrated, the relative change in the probability that transcription wins the race can be much sharper, than the relative change in the ribosome's speed. Our work confirms that the *hyper-sensitivity* of attenuation *depends on the number of translation versus transcription steps*.

Our rule-based model is compact and mathematically well-founded. We believe that it is easier to understand than a stochastic model defined by means of probability density functions or the chemical master equation. As we plan to demonstrate in future work, it is easier to extend by additional concurrent issues, that the master equation approach can not well explicitly render.

The following initial solution allows us to simulate the hyper-sensitivity of attenuation, disregarding its control by secondary mRNA structures:

$$\text{RNAP} \cdot \text{DNA}_0 \, , \, \oplus_{i=1}^{n-1} \text{DNA}_i \, , \, \text{Ribosome} \cdot \text{mRNA}_0 \, , \, \oplus_{i=1}^{m-1} \text{mRNA}_i$$

In what regards notational conventions, we use molecule names $\mathcal{N} = \{\text{RNAP, mRNA, DNA, Ribosome}\}$, attribute values $\mathcal{C} = \mathbb{N}_0$, function symbols $\mathcal{F} = \{+\}$, variables $\mathcal{V} = \{i\}$, and value parameters $n, m \in \mathcal{C}$. Because our model's attributed molecules bear only few arguments, for the sake of presentation we slightly differ from the formal syntax introduced in Section 3. We write attributes as indices for molecule names, instead of parenthesizing them. For instance we write DNA_i instead of $\text{DNA}(i)$, and $\oplus_{i=1}^{n-1} \text{DNA}_i$ instead of $\text{DNA}(1), \ldots, \text{DNA}(n-1)$.

Our model consists in the following rule schemas:

$$\forall i \in \{0,\ldots,n-2\}.\text{RNAP} \cdot \text{DNA}_i \, , \, \text{DNA}_{i+1} \rightarrow_{e_1} \text{DNA}_i \, , \, \text{RNAP} \cdot \text{DNA}_{i+1} \tag{2}$$

$$\text{RNAP} \cdot \text{DNA}_{n-1} \rightarrow_{e_1} \text{RNAP} \, , \, \text{DNA}_{n-1} \tag{3}$$

$$\forall i \in \{0,\ldots,m-2\}.\text{Ribosome} \cdot \text{mRNA} \, , \, \text{mRNA}_{i+1} \rightarrow_{e_2} \text{mRNA}_i \, , \, \text{Ribosome} \cdot \text{mRNA}_{i+1} \tag{4}$$

$$\text{Ribosome} \cdot \text{mRNA}_{m-1} \rightarrow_{e_2} \text{Ribosome} \, , \, \text{mRNA}_{m-1} \tag{5}$$

Two rules represent elongation by the polymerase and the ribosome: the $n-1$ transcription steps apply the rate constant e_1 in (2), and the $m-1$ translation steps in (4) use the rate constant e_2. The two remaining rules model termination in a simplistic manner. RNAP's dissociation is the last of its n steps (3), and that of the ribosome the last of m steps (5). Note that our dissociation rules re-use the elongation rate constants, which is not quantitatively realistic, but a useful coding trick. It allows us to agree, regarding the number of simulated steps, with Elf and Ehrenberg (2005), whose hyper-sensitivity model disregards dissociation. Indeed, using the same rate constants, we reproduce exactly their Figure 2.

The plot of our simulation results (Figure 3, right) is organized as follows. The y-axis gives the probability that RNAP wins the race, i.e. a value between zero and one, that indicates the proportion of simulations in which RNAP dissociates first. The x-axis reports the translation rate on a logarithmic scale, that that we vary from 0.01 to 100 codons per second in our simulations. We report simulation results for three models, each of which delivers one curve in Figure 3. These models differ in their numbers of translation (m) versus transcription (n) steps: we investigate the combinations of ($m=1$, $n=1$), ($m=1$, $n=50$), and ($m=10$, $n=50$). Simulation was carried out with the Kappa Factory (Danos et al. 2007).

Let us compare the sensitivity of the three different models. When $m=1$ and $n=1$, the probability curve decreases gently, and already shows a non-linear behavior. Increasing the number of transcription steps to $n=50$ steepens the curve, i.e. augments the sensitivity. The transition becomes even sharper when the number of translation steps reaches high values ($m=10$, $n=50$). Such values hold for systems where, unlike at *E.coli*'s *trp* operon, attenuation is the sole control mechanism (Kasai 1974).

We adopt the calculation of rate constants from Elf and Ehrenberg (2005), to ensure comparability of the outcomes of the three races. We keep the total time to perform the whole series of n transcription steps constant, such that $1/e_1' = 1\text{sec}$. Thus, the rate constant for one individual transcription step out of n is $e_1 = n \cdot e_1'$. For one translation step the rate constant is $e_2 = m \cdot e_2'$. Hereby e_2' is the average rate for m translation steps, which varies logarithmically between 0.01 and 100.

5 MODELING TRANSCRIPTIONAL ATTENUATION

This section introduces our rule-based model of ribosome-mediated transcriptional attenuation at *E.coli*'s tryptophan operon. It refines Section 4's simplistic model in several points. We dynamically add new mRNA segments during transcription, when the RNAP transcribes the DNA position corresponding to their last codon. mRNA segments pair into hairpins, unless they are temporarily covered by the ribosome's footprint. Hairpins play their respective regulatory roles on transcription, and are disrupted by the advancing ribosome. Last but not least, our model switches between two abstraction levels for the first mRNA segment: S_1 is either represented as a whole, or by an explicit sequence of codons. In Section 5.2, we present simulation results obtained from our model, and validate them against the work of Elf and Ehrenberg (2005).

5.1 Rule Schemas

The rule schemas of our detailed attenuation model rely on the notational conventions from Section 4. We use molecule names $\mathscr{N} = \{\text{RNAP}, \text{mRNA}, \text{DNA}, \text{Ribosome}, \text{S}\}$, attribute values $\mathscr{C} = \mathbb{N}_0 \cup \{\text{fr}, \text{hp}, \text{bl}\}$, function symbols $\mathscr{F} = \{+\}$, and variables $\mathscr{V} = \{i,n,m,t,x\}$. Molecules with two attributes S_i^x represent segments of the mRNA leader, where the lower index denotes the segment's number ($i \in \{1,2,3,4\}$), and the upper index x its state:

- *hairpin* (hp): paired with a neighboring segment into a hairpin (e.g. the pause hairpin $S_1^{\text{hp}} \cdot S_2^{\text{hp}}$),
- *blocked* (bl): masked by the ribosome's footprint (only for S_2),
- *free* (fr): available for hairpin formation.

Our model's initial solution reflects the starting configuration for the attenuation race, as illustrated in Figure 1 (A):

$$\text{RNAP} \cdot \text{DNA}_0 \, , \, \oplus_{i=1}^{50} \text{DNA}_i \, , \, \text{Ribosome} \cdot \text{mRNA}_6 \, , \, \text{mRNA}_7 \, , \, \text{mRNA}_8 \, , \, \text{mRNA}_9 \, , \, \text{mRNA}_{15} \, , \, S_1^{\text{hp}} \cdot S_2^{\text{hp}} \tag{6}$$

$$\forall i \in \{1,2,3\}.S_i^{fr}, S_{i+1}^{fr} \quad \rightarrow_h \quad S_i^{hp} \cdot S_{i+1}^{hp} \tag{7}$$

$$\forall m \in \{7,8,9,13,14\}.\text{Ribosome} \cdot \text{mRNA}_m, \text{mRNA}_{m+1} \quad \rightarrow_{e_2} \quad \text{mRNA}_m, \text{Ribosome} \cdot \text{mRNA}_{m+1} \tag{8}$$

$$\forall t \in \{10,11\}.\text{Ribosome} \cdot \text{mRNA}_t, \text{mRNA}_{t+1} \quad \rightarrow_{e_3} \quad \text{mRNA}_t, \text{Ribosome} \cdot \text{mRNA}_{t+1} \tag{9}$$

$$\text{Ribosome} \cdot \text{mRNA}_6, \text{mRNA}_7, S_1^{hp} \cdot S_2^{hp} \quad \rightarrow_{e_2} \quad \text{mRNA}_6, \text{Ribosome} \cdot \text{mRNA}_7, \oplus_{i=10}^{14}\text{mRNA}_i, S_2^{fr} \tag{10}$$

$$\text{Ribosome} \cdot \text{mRNA}_{12}, \text{mRNA}_{13}, S_2^{fr} \quad \rightarrow_{e_2} \quad \text{mRNA}_{12}, \text{Ribosome} \cdot \text{mRNA}_{13}, S_2^{bl} \tag{11}$$

$$\text{Ribosome} \cdot \text{mRNA}_{12}, \text{mRNA}_{13}, S_2^{hp} \cdot S_3^{hp} \quad \rightarrow_{e_2} \quad \text{mRNA}_{12}, \text{Ribosome} \cdot \text{mRNA}_{13}, S_2^{bl}, S_3^{fr} \tag{12}$$

$$\text{Ribosome} \cdot \text{mRNA}_{15}, \oplus_{i=10}^{14}\text{mRNA}_i, S_2^{bl} \quad \rightarrow_d \quad \text{Ribosome}, \text{mRNA}_{15}, S_1^{fr}, S_2^{fr} \tag{13}$$

$$\forall n \in \{1,\ldots,49\} \setminus \{35,46,47\}.\text{RNAP} \cdot \text{DNA}_n, \text{DNA}_{n+1} \quad \rightarrow_{e_1} \quad \text{DNA}_n, \text{RNAP} \cdot \text{DNA}_{n+1} \tag{14}$$

$$\forall x \in \{fr,bl\}.\text{RNAP} \cdot \text{DNA}_0, \text{DNA}_1, S_2^x \quad \rightarrow_{e_1} \quad \text{DNA}_0, \text{RNAP} \cdot \text{DNA}_1, S_2^x \tag{15}$$

$$\text{RNAP} \cdot \text{DNA}_{35}, \text{DNA}_{36} \quad \rightarrow_{e_1} \quad \text{DNA}_{35}, \text{RNAP} \cdot \text{DNA}_{36}, S_3^{fr} \tag{16}$$

$$\text{RNAP} \cdot \text{DNA}_{46}, \text{DNA}_{47} \quad \rightarrow_{e_1} \quad \text{DNA}_{46}, \text{RNAP} \cdot \text{DNA}_{47}, S_4^{fr} \tag{17}$$

$$\text{RNAP} \cdot \text{DNA}_{47}, S_3^{hp} \cdot S_4^{hp} \quad \rightarrow_{e_1} \quad \text{DNA}_{47}, \text{RNAP}, S_3^{hp} \cdot S_4^{hp} \tag{18}$$

$$\text{RNAP} \cdot \text{DNA}_{47}, \text{DNA}_{48}, S_4^{fr} \quad \rightarrow_{e_1} \quad \text{DNA}_{47}, \text{RNAP} \cdot \text{DNA}_{48}, S_4^{fr} \tag{19}$$

The RNAP has transcribed the leader mRNA up to and including S_1 and S_2 (that are paired into the pause hairpin), and is paused on the zero-th DNA nucleotide (of a series of 50). The ribosome has initiated translation and is located on the 6th codon of the leader transcript. Note that we chose not to render the codons preceding 6, since they do not matter to the attenuation race, nor the rules for the initiation of transcription and translation. We explicitly render the mRNA codons 6 to 9, which precede the segment S_1, and the stop codon 15, that is located between the segments S_1 and S_2.

Let us now explain our abstraction level switching regarding the first mRNA segment. Recall that S_1 comprises the mRNA codons 10 to 14, as illustrated in Figure 1 (B). While the ribosome is advancing step by step over these, we explicitly represent $\oplus_{i=10}^{14}\text{mRNA}_i$. This implicitly sequesters the first segment, that is covered by the ribosome's footprint while being translated, from hairpin formation. When however the ribosome is not processing the *trp* control between mRNA_7 and the stop codon mRNA_{15}, we represent S_1 as a whole. This allows formation of the pause hairpin, provided S_2 is available.

Let us now discuss our model of *translation*, rule schemas (8) to (13). The bulk of translation steps, that do not have side effects, do not depend on tryptophan availability or other side conditions, are covered by (8). They bear the reaction rate constant $e_2 = 15s^{-1}$. For the control codons 10 and 11 in rule (9), we apply the distinct elongation rate constant e_3, that varies within $]0,15]s^{-1}$ in our simulations. Exceptions to these two generic rules are when the ribosome is located at codons 6 (the next step melts the pause hairpin), 12 (the next step blocks S_2), and 15 (from where the ribosome eventually dissociates).

We now consider the detailed dynamics starting from our initial solution (6). The ribosome is about to step from mRNA_6 to mRNA_7, disrupting the pause hairpin in the application of rule (10). The rule's right hand side shows two side effects. On one hand, we switch down from the abstraction of the segment S_1 as a whole, to the explicit representation of the codons $\oplus_{i=10}^{14}\text{mRNA}_i$ that constitute it. On the other, S_2 switches into its *free* state. Both effects are reverted later upon dissociation of the ribosome from the stop codon mRNA_{15} in rule (13), where codons $\oplus_{i=10}^{14}\text{mRNA}_i$ are replaced by segment S_1^{fr}.

We distinguish two rules for the ribosome's step from mRNA_{12} to mRNA_{13}, after which the second segment is obscured by the ribosome's footprint, and thus unavailable for hairpin formation. Rule (11) applies when S_2 is initially free. In its counterpart rule (12), S_2 is initially paired into the anti-terminator hairpin, that the ribosome's advance melts. The henceforth blocked second segment (S_2^{bl}) is freed when the ribosome dissociates from the stop codon, which rule (13) represents.

Hairpin formation is covered by rule schema (7), be it for the pause hairpin, the anti-terminator or the terminator. We assign it a rate constant $h = 10000$, where the precise numeric value is of lesser importance. What matters is that hairpin formation occurs on a faster time scale than any other reaction.

We last present the transcription rules (14) to (19). Rule schema 14 represents the general case of a transcription step, as it did in Section 4. It applies to all DNA positions from 0 to 50 with the following exceptions that we discuss in the order they are applied, starting from our initial solution. Transcription *resumes* at position DNA_0 after the pause hairpin has been disrupted in (15). It applies when S_2 is either free or blocked. Note that it may appear simpler to check the *absence* of the pause hairpin, but unfortunately such negative tests are not supported by current rule-based frameworks.

The remaining rules deal with the creation of new segments of mRNA, and (anti)termination of transcription as RNAP reaches
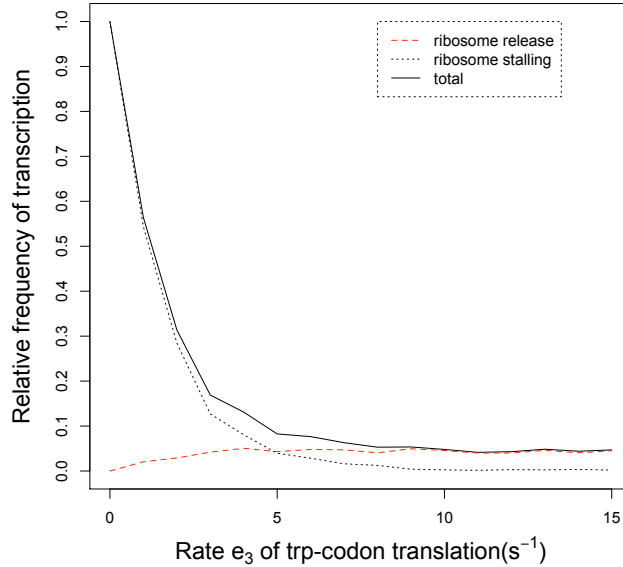
Figure 4: Relative frequency of transcription as a function of the rate of the Trp codon translation in the leader mRNA, distinguishing between anti-terminator formation during ribosome stalling, and after ribosome release from the stop codon.

the end of the control region on DNA. As RNAP steps over to DNA_{36}, the RNA segment S_3 is spawn by rule (16). S_4 follows at DNA_{47} by rule (17). Transcription terminates on DNA_{47} provided there is a terminator hairpin, as witnessed by the presence of S_4^{hp} in reaction (18). If conversely S_4 remains free, this indicates the absence of the terminator hairpin, hence transcription proceeds over DNA_{48} rule (19) and into the operon.

Finally we summarize the use of n-ary rules, distinguishing three categories of rules with more than two reactants in our models. Such n-ary rules are not representable in an intelligible manner within object-centric approaches limited to binary interactions, namely π-calculus based modeling languages. In the *first category*, we check whether the current solution fulfills a certain *prerequisite*, e.g. contains a certain molecule, or a certain molecule in a specific state. It might appear more convenient to impose negative conditions on rule application, but this is neither supported by our language, nor other current rule-based frameworks. For instance, rule 15 resumes transcription if the pause hairpin is absent, i.e. if S_2's state is free or blocked. Similarly, the rules for abortion versus continuation of transcription (rules 18 and 19) depend on the terminator hairpin. The *second category* are binary reactions entailing the *modification* of a third agent. Examples are the rules for blocking S_2 as the ribosome proceeds to $mRNA_{13}$, possibly disrupting the anti-terminator hairpin (rules 11 and 12). The *third category* is the previously discussed abstraction level switching, e.g. re-assembling S_1 from the codons 10 to 14 in rule 13.

5.2 Simulation

Figure 4 shows the dependency of the relative transcription frequency on the rate of *trp*-codon translation. We varied e_3 from 0 to 15 s^{-1} in steps of one, performed 5000 Gillespie simulations per value within the Kappa factory. Two distinct pathways lead to the anti-terminator hairpin, that the figure distinguishes. The predominant pathway at low rate of trp-codon translation is that $S_2 \cdot S_3$ forms while the ribosome is stalled on the control codons. It becomes rarer when increasing the rate, as shown by the curve *ribosome stalling*.

In a second pathway, the anti-terminator hairpin forms after the ribosome has released from the stop codon 15. Although rarer, this occasionally occurs and represents the basal read-through level of the trp operon (see Section 2). The curve *ribosome release* increases with the rate of trp-codon translation.

Our results shown in Figure 4 confirm those of Elf and Ehrenberg (2005). However, even if the curves have the same shape, the asymptotic decrease of *ribosome stalling* toward 0 is less sharp in our case. While in Elf and Ehrenberg (2005) the curves for ribosome release and ribosome stalling cross at a rate of $6s^{-1}$, in ours they already do so at $5s^{-1}$. Moreover our experiments predict a rate of basal transcription of slightly under 5% when *trp*-codon translation is efficient, where Elf and Ehrenberg predict 8%. Both differences still deliver open questions.

6   CONCLUSION

We have shown that rule-based modeling provides concise and elegant models for the fine-grained mechanism of transcriptional attenuation, a problem left open by previous work on discrete event modeling of the tryptophan operon (Simão et al. 2005). The core ingredient for our model are rule schemas and n-ary chemical reactions. The need for n-ary reactions renders representations in object-centered languages such as the stochastic pi-calculus (Regev 2002, Phillips and Cardelli 2007, Kuttler et al. 2007) inappropriate, in practice. We used the Kappa factory for stochastic simulation (Danos et al. 2007), which provides convenient analysis tools.

As modeling technique, we identified nodes of DNA and RNA sequences by numbers, and addressed successors by simple arithmetic. This technique has its limitations when polymers become more complex than simple lists. Alternatively, we could assign names to molecular domains, and memorize those names in attribute values of adjacent molecules, similarly to Kappa. As shown in Section 3.4, the cost for such an alternative is that meaningless instances of rule schemas may be generated.

In future work, we plan a more detailed analysis of our quantitative predictions for the *trp* operon, to simulate the dynamics of the *his* operon (that differs in its control region architecture, see Kasai (1974)), and to investigate how rule-based modeling could cover transcriptional attenuation mediated by other agents than the ribosome (Gollnick et al. 2005, Gutierrez-Preciado et al. 2005, Yanofsky 2007). We also wish to elucidate quantitative differences between our predictions and those of Elf and Ehrenberg (2005), regarding the basal read-through level. Their model is much simpler than ours, since attenuation is reduced to a race between the ribosome and the RNA polymerase, based on their relative position on two independent strands of mRNA and DNA respectively. We suppose that our model with explicit hairpins and creation of mRNA segments during transcription includes some behaviors not covered by Elf and Ehrenberg (2005).

REFERENCES

Arkin, A., J. Ross, and H. H. McAdams. 1998. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells. *Genetics* 149:1633–1648.

Barboric, M., and B. M. Peterlin. 2005. A new paradigm in eukaryotic biology: HIV Tat and the control of transcriptional elongation. *PLoS Biology* 3 (2): 0200–02003.

Beisel, C. L., and C. D. Smolke. 2009, 04. Design principles for riboswitch function. *PLoS Computational Biology* 5 (4): e1000363.

Cardelli, L., and G. Zavattaro. 2008. On the computational power of biochemistry. In *Proceedings of the 3rd international conference on Algebraic Biology*, 65–80. Berlin, Heidelberg: Springer-Verlag.

Chabrier-Rivier, N., F. Fages, and S. Soliman. 2005. The biochemical abstract machine BioCham. In *Proceedings of CMSB 2004*, Volume 3082 of *Lecture Notes in Bioinformatics*, 172–191.

Ciocchetta, F., and J. Hillston. 2008. Bio-PEPA: a framework for modelling and analysis of biological systems. *Theoretical Computer Science*. To appear.

Danos, V., J. Feret, W. Fontana, R. Harmer, and J. Krivine. 2007. Rule-based modelling of cellular signalling. In *18th International Conference on Concurrency Theory*, Volume 4703 of *Lecture Notes in Computer Science*, 17–41.

Danos, V., J. Feret, W. Fontana, and J. Krivine. 2007. Scalable simulation of cellular signaling networks. In *5th Asian Symposium on Programming Languages and Systems*, Volume 4807 of *Lecture Notes in Computer Science*, 139–157.

Dematté, L., C. Priami, and A. Romanel. 2008. The beta workbench: A tool to study the dynamics of biological systems. *Briefings in Bioinformatics* 9 (5): 437–449.

Elf, J., and M. Ehrenberg. 2005. What makes ribosome-mediated trascriptional attenuation sensitive to amino acid limitation? *PLoS Computational Biology* 1 (1): 14–23.

Gillespie, D. T. 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics* 22:403–434.

Gollnick, P. 2004. Trp operon and attenuation. In *Encyclopedia of Biological Chemistry*, ed. W. J. Lennarz and M. D. Lane, 267 – 271. New York: Elsevier.

Gollnick, P., P. Babitzke, A. Antson, and C. Yanofsky. 2005. Complexity in regulation of tryptophan biosynthesis in Bacillus subtilis. *Annual Review of Genetics* 39 (1): 47–68.

Gutierrez-Preciado, A., R. Jensen, C. Yanofsky, and E. Merino. 2005. New insights into regulation of the tryptophan biosynthetic operon in Gram-positive bacteria. *Trends in Genetics* 21 (8): 432–436.

Hlavacek, W. S., J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana. 2006. Rules for modeling signal-transduction systems. *Science STKE* re6:1–18.

JR Roesser, Y. N., and C. Yanofsky. 1989. Regulation of basal level expression of the tryptophan operon of Escherichia coli. *J Biol Chem* 264 (21): 12284–8.

Kasai, T. 1974. Regulation of the expression of the histidine operon in Salmonella typhimurium. *Nature* 249:523–527.

Konan, K. V., and C. Yanofsky. 1999. Role of ribosome release in regulation of tna operon expression in Escherichia coli. *J. Bacteriol.* 181:1530–1536.

Krivine, J., R. Milner, and A. Troina. 2008. Stochastic bigraphs. In *24th Conference on the Mathematical Foundations of Programming Semantics*, Volume 218 of *Electronical notes in theoretical computer science*, 73–96: Elsevier.

Kuttler, C., C. Lhoussaine, and J. Niehren. 2007, July. A stochastic pi calculus for concurrent objects. In *Second International Conference on Algebraic Biology*, Volume 4545 of *Lecture Notes in Computer Science*, 232–246: Springer Verlag.

Phillips, A., and L. Cardelli. 2007. Efficient, correct simulation of biological processes in the stochastic pi-calculus. In *Computational Methods in Systems Biology, International Conference*, Volume 4695 of *Lecture Notes in Computer Science*, 184–199: Springer Verlag.

Pradalier, S., A. Credi, M. Garavelli, C. Laneve, and G. Zavattaro. 2007. Modelization and simulation of nano devices in the nano-kappa calculus. In *Computational Methods in Systems Biology, International Conference CMSB 2007*.

Ramsey, S., D. Orrell, and H. Bolouri. 2005. Dizzy: stochastic simulation of large-scale genetic regulatory networks. *Journal of Bioinformatics and Computational Biology* 3 (2): 415–436.

Regev, A. 2002. *Computational systems biology: A calculus for biomolecular knowledge*. Tel Aviv University. PhD thesis.

Santillan, M., and E. S. Zeron. 2004. Dynamic influence of feedback enzyme inhibition and transcription attenuation on the tryptophan operon response to nutritional shifts. *Journal of Theoretical Biology* 231 (2): 287–298.

Simão, E., E. Remy, D. Thieffry, and C. Chaouiya. 2005. Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in E .coli. In *ECCB/JBI*, 190–196.

Trun, N., and J. Trempy. 2003. *Fundamental bacterial genetics*, Chapter Gene expression and regulation, 191–212. Blackwell.

von Heijne, G., L. Nilsson, and C. Blomberg. 1977. Translation and messenger RNA secondary structure. *Journal of Theoretical Biology* 68:321–329.

Yanofsky, C. 1981. Attenuation in the control of expression of bacterial operons. *Nature* 289:751–758.

Yanofsky, C. 2000. Transcription attenuation: once viewed as a novel regulatory strategy. *J Bacteriology* 182 (1): 1–8.

Yanofsky, C. 2007. RNA-based regulation of genes of tryptophan synthesis and degradation, in bacteria. *RNA - A publication of the RNA Society* 13 (8): 1141–1154.

AUTHOR BIOGRAPHIES

CEDRIC LHOUSSAINE is assistant professor at Lille1 University and head of the bioComputing team in the LIFL lab (`http://www2.lifl.fr/BioComputing`), and reacheable by email at `cedric.lhoussaine@lifl.fr`.

CELINE KUTTLER is assistant professor within the bioComputing team. Contact: `celine.kuttler@lifl.fr`

MIRABELLE NEBUT is assistant professor within the bioComputing team. Contact: `mirabelle.nebut@lifl.fr`